CLAIMS

[0051] What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A circuit having support for normalization of significands, comprising:

a first register block, said first register block including at least one first register for holding a first exponent and a first significand of a first floating point number and a first logic capable of left shifting the significand of the first floating point number;

a second register block, said second register block including at least one second register for holding a second exponent and a second significand of a second floating point number;

a plurality of flags coupled to said first register block, each of said plurality of flags having a state based on the contents of said first significand;

an arithmetic logic unit coupled to said first register block, said second register block, and said plurality of flags, said arithmetic logic unit causing the first logic to left shift the first significand based upon the states of said plurality of flags.

2. The circuit of claim 1, wherein said plurality of flags further comprises:

an I^{th} flag, wherein I is a non-negative integer, said I^{th} flag which is set to a first sate when the 2^I most significant bits of said first significand are each zeros and a second state if any of the 2^I most significant bits is non-zero.

3. The circuit of claim 2, wherein said arithmetic logic unit causes said first logic to left shift by 2^I bits the first significand if said I^{th} flag is set to the first state.

4. The circuit of claim 3, wherein said arithmetic logic unit is coupled to a temporary storage location for storing an adjustment to be subtracted from said first exponent, and increments said adjustment by 2^I if said first flag is set to the first state.

5. The circuit of claim 2, wherein I is 0.

6. The circuit of claim 2, wherein I is 1.

7. The circuit of claim 2, wherein I is 2.

8. The circuit of claim 2, wherein I is 3.

9. The circuit of claim 1, wherein said arithmetic logic unit is coupled to a temporary storage location.

10. The circuit of claim 9, wherein said temporary storage location is a register in a register file.

11. The circuit of claim 9, wherein said temporary storage location is a main memory accessed through a memory interface.

12. The circuit of claim 1, wherein:

said plurality of flags further comprises,

an I^{th} flag, wherein I is a positive integer of at least 3, which is set to a first state when the 2^I most significant bits of said first significand are each zeros and a second state if any of the 2^I most significant bits of said first significand is non-zero;

an $(I-1)^{th}$ flag which is set to a first state when the $2^{(I-1)}$ most significant bits of said first significand are each zeros and a second state if any of the $2^{(I-1)}$ most significant bits of said first significand is non-zero;

an $(I-2)^{th}$ flag which is set to a first state when the $2^{(I-2)}$ most significant bits of said first significand are each zeros and a second state if any of the $2^{(I-2)}$ most significant bits of said first significand is non-zero; and

an $(I-3)^{th}$ flag which is set to a first state when the $2^{(I-3)}$ significant bits of said first significand are each zeros and a second state if the $2^{(I-3)}$ significant bits of said first significand is non-zero; and wherein

said arithmetic logic unit is coupled to a temporary storage location, said arithmetic logic unit initially setting the temporary storage location to zero, then modifying said temporary location based upon the state of the plurality of flags, and finally modifying said first exponent based on the contents of said temporary location.

13. The circuit of claim 12, wherein said temporary storage location is a register in a register file.

14. The circuit of claim 12, wherein said temporary storage location is a main memory accessed through a memory interface.

15. The circuit of claim 12 wherein said arithmetic logic unit modifies the first exponent by subtracting the contents of said temporary location from said first exponent.

16. The circuit of claim 12, wherein I is equal to 3.

17. A massively parallel processing system, comprising:

a main memory;

an array of processing elements, each processing element of the array being coupled to said main memory and other processing elements of said array, wherein each of said processing elements comprises,

a first register block, said first register block including at least one first register for holding a first exponent and a first significand of a first floating point number and a first logic capable of left shifting the significand of the first floating point number;

a second register block, said second register block including at least one second register for holding a second exponent and a second significand of a second floating point number;

a plurality of flags coupled to said first register block, each of said plurality of flags having a state based on the contents of said first significand;

an arithmetic logic unit coupled to said first register block, said second register block, and said plurality of flags, said arithmetic logic unit causing the first logic to left shift the first significand based upon the states of said plurality of flags.

18. The massively parallel processing system of claim 17, wherein said plurality of flags further comprises:

an I^{th} flag, wherein I is a non-negative integer, said I^{th} flag which is set to a first sate when the 2^I most significant bits of said first significand are each zeros and a second state if any of the 2^I most significant bits is non-zero.

19. The massively parallel processing system of claim 18, wherein said arithmetic logic unit causes said first logic to left shift by 2^I bits the first significand if said I^{th} flag is set to the first state.

20. The massively parallel processing system of claim 19, wherein said arithmetic logic unit is coupled to a temporary storage location for storing an adjustment to be subtracted from said first exponent, and increments said adjustment by 2^I if said first flag is set to the first state.

21. The massively parallel processing system of claim 18, wherein I is 0.

22. The massively parallel processing system of claim 18, wherein I is 1.

23. The massively parallel processing system of claim 18, wherein I is 2.

24. The massively parallel processing system of claim 18, wherein I is 3.

25. The massively parallel processing system of claim 17, wherein said arithmetic logic unit is coupled to a temporary storage location.

26. The massively parallel processing system of claim 25, wherein said temporary storage location is a register in a register file.

27. The massively parallel processing system of claim 25, wherein said temporary storage location is a main memory accessed through a memory interface.

28. The massively parallel processing system of claim 17, wherein:

said plurality of flags further comprises,

an I^{th} flag, wherein I is a positive integer of at least 3, which is set to a first state when the 2^I most significant bits of said first significand are each zeros and a second state if any of the 2^I most significant bits of said first significand is non-zero;

a $(I-1)^{th}$ flag which is set to a first state when the $2^{(I-1)}$ most significant bits of said first significand are each zeros and a second state if any of the $2^{(I-1)}$ most significant bits of said first significand is non-zero;

a $(I-2)^{th}$ flag which is set to a first state when the $2^{(I-2)}$ most significant bits of said first significand are each zeros and a second state if any of the $2^{(I-2)}$ most significant bits of said first significand is non-zero; and

a $(I-3)^{th}$ flag which is set to a first state when the $2^{(I-3)}$ most significant bits of said first significand are each zeros and a second state if the $2^{(I-3)}$ significant bits of said first significand is non-zero; and wherein

said arithmetic logic unit is coupled to a temporary storage location, said arithmetic logic unit initially setting the temporary storage location to zero, then modifying said temporary location based upon the state of the plurality of flags, and finally modifying said first exponent based on the contents of said temporary location.

29. The massively parallel processing system of claim 28, wherein said temporary storage location is a register in a register file.

30. The massively parallel processing system of claim 28, wherein said temporary storage location is a main memory accessed through a memory interface.

31. The massively parallel processing system of claim 17 wherein said arithmetic logic unit modifies the first exponent by subtracting the contents of said temporary location from said first exponent.

32. The massively parallel processing system of claim 18, wherein I is equal to 3.

33. A method for normalizing the significand of a floating point number stored in a processing element having an exponent register, a plurality of significand registers, an I^{th} flag indicating whether the 2^I most significant bits of the significand are each zero, a $(I-1)^{th}$ flag indicating whether the 2^I most significant bits of the significand are each zero, a $(I-2)^{th}$ flag indicating whether the $2^{(I-2)}$ most significant bits of the significand are each zero, a $(I-3)^{th}$ flag indicating whether the $2^{(I-3)}$ most significant bit of the significand is zero, and a temporary variable, wherein I is an integer of at least 3, said method comprising the step of:

(a) initializing the temporary variable to zero;

(b) if said I^{th} flag is set, left shifting the significand by 2^I bits and incrementing the temporary variable by 2^I;

(c) if said $(I-1)^{th}$ flag is set, left shifting the significand by $2^{(I-1)}$ bits and incrementing the temporary variable by $2^{(I-1)}$;

(d) if said $(I-2)^{th}$ flag is set, left shifting the significand by $2^{(I-2)}$ bits and incrementing the temporary variable by $2^{(I-2)}$;

(e) if said $(I-3)^{th}$ flag is set, left shifting the significand by $2^{(I+3)}$ bit and incrementing the temporary variable by $2^{(I-3)}$; and

(f) decrementing the exponent register by the value of the temporary variable.

34. The method of claim 33, wherein I is equal to 3.

35. The method of claim 33, wherein step (a) is performed before step (b).

36. The method of claim 43, wherein step (c) is performed after step (b).

37. The method of claim 44, wherein step (d) is performed after step (c).

38. The method of claim 45, wherein step (e) is performed after step (d).

39. The method of claim 46, wherein step (f) is performed after step (e).